Trabajo Práctico 2: Git y GitHub

Alumna: Karen Yanet Guardia
Materia: Programación I

Objetivo:

El estudiante desarrollará competencias para trabajar con Git y GitHub, aplicando conceptos fundamentales de control de versiones, colaboración en proyectos y resolución de conflictos, en un entorno simulado y guiado.

Resultados de aprendizaje:

1. Comprender los conceptos básicos de Git y GitHub: Identificar y explicar los principales términos y procesos asociados con Git y GitHub, como repositorios, ramas, commits, forks, etiquetas y repositorios remotos.

2. Manejar comandos esenciales de Git: Ejecutar comandos básicos para crear, modificar, fusionar y gestionar ramas, commits y repositorios, tanto en local como en remoto.

3. Aplicar técnicas de colaboración en GitHub: Configurar y utilizar repositorios remotos, realizar forks, y gestionar pull requests para facilitar el trabajo colaborativo.

4. Resolver conflictos en un entorno de control de versiones: Identificar, analizar y solucionar conflictos de merge generados en un flujo de trabajo con múltiples ramas.

Actividades

1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :

• ¿Qué es GitHub?

GitHub es una plataforma de alojamiento de código fuente que utiliza Git, un sistema de control de versiones. Es una comunidad donde nosotros los desarrolladores podemos compartir nuestros repositorios de forma pública o privada de manera remota, y podemos colaborar con otros desarrolladores, hacer seguimiento de cambios y mantener un historial completo del desarrollo de los proyectos.

• ¿Cómo crear un repositorio en GitHub?

1) Inicias sesión en GitHub.
2) Haces click en el botón + que despliega opciones y haces click en “New repository”.
3) Rellenas el formulario con el nombre del repositorio, una descripción opcional y si lo dejamos en público o privado.
4) Finalmente le das click al botón verde de “Create repository”.

• ***¿Cómo crear una rama en Git?***

Se utiliza el comando: git branch nombre_de_la_rama

Este comando creará una nueva rama llamada nombre_de_la_rama

• ***¿Cómo cambiar a una rama en Git?***

Para cambiar de una rama a otra se utiliza el comando: git checkout nombre_de_la_rama

Y en versiones más recientes de git puedes usar este comando: git switch nombre_de_la_rama

• ***¿Cómo fusionar ramas en Git?***

Para fusionar ramas en Git usamos el comando: git merge nombre_de_la_rama

Ese comando fusionará los cambios de nombre_de_la_rama a la rama en la que te encuentres actualmente (por ejemplo si estabas en la rama main irán a esa rama los cambios).

• ***¿Cómo crear un commit en Git?***

Un commit es un registro de los cambios realizados en los archivos del proyecto.

Así que por eso, primero nos aseguramos de haber hecho cambios para agregar al staging area, que es el paso previo a hacer un commit.

Esto lo hacemos con el comando git add .

Seguido a eso, hacemos el comando git commit -m “Mensaje descriptivo del commit”. Este comando es el punto que guarda el cambio que se realizó.

• ***¿Cómo enviar un commit a GitHub?***

Después de haber realizado el commit en el repositorio local, si queremos que también se vea reflejado en el repositorio remoto tenemos que usar el siguiente comando:

git push -u origin main -> Graba el cambio en la branch main del repositorio remoto.

• ***¿Qué es un repositorio remoto?***

Un repositorio es un espacio donde se guarda todo el contenido de un proyecto, incluyendo el código, archivos, y el historial de cambios. El repositorio remoto es un repositorio que está alojado en internet o en una red privada, es decir en un servidor externo. Permite guardar el código en la nube y compartirlo con otros para trabajar en equipo. Algunas plataformas que permite alojar repositorios remotos son: GitHub, GitLab o Bitbucket.

• ***¿Cómo agregar un repositorio remoto a Git?***

Para agregar un repositorio remoto se utiliza el siguiente comando:

git remote add origin URL-del-repositorio

• ***¿Cómo empujar cambios a un repositorio remoto?***

Para enviar los cambios realizados en el repositorio local al repositorio remoto se usa el comando:

git push origin nombre_de_la_rama (en la que se realizó los cambios).

• ¿Cómo tirar de cambios de un repositorio remoto?

Para descargar los cambios desde el repositorio remoto y fusionarlos con el repositorio local se usa el comando:

git pull origin nombre_de_la_rama

• ¿Qué es un fork de repositorio?

Un fork de repositorio es una copia de un repositorio remoto que se guarda en tu propia cuenta de GitHub para poder hacer cambios sin modificar el repositorio original.

• ¿Cómo crear un fork de un repositorio?

1) Iniciamos sesión en GitHub.
2) Ingresamos al repositorio remoto que queremos copiar.
3) Hacemos click en el botón fork y esto creará una copia del repositorio en nuestra cuenta de GitHub.

• ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

1) Después de hacer cambios en mi fork y empujarlos, voy a ir a la página del repositorio original.
2) GitHub me sugerirá abrir un “pull request”.
3) Hago click en “New Pull Request” y elijo mi rama y los cambios que deseo enviar.

• ¿Cómo aceptar una solicitud de extracción?

Si sos el propietario del repositorio, podes aceptar el pull request haciendo click en “Merge Pull Request” y luego en “Confirm Merge”.

• ¿Qué es una etiqueta en Git?

Una etiqueta (tag) en Git es una referencia que marca un punto específico en la historia del proyecto, como una versión estable. Se usa, por ejemplo, para indicar lanzamientos o versiones importantes.

• ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta en Git se usa el comando:

git tag -a nombre_etiqueta -m “Mensaje descriptivo de la etiqueta”

• ¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta a GitHub usamos el comando:

git push origin nombre_etiqueta

• ¿Qué es un historial de Git?

El historial de Git es el registro de todos los cambios realizados en el repositorio del proyecto, incluyendo los commits, fechas, autores y mensajes. Permite ver cómo fue evolucionando el proyecto.

• ¿Cómo ver el historial de Git?

Para poder ver el historial de Git se utiliza el siguiente comando:

git log

• ¿Cómo buscar en el historial de Git?

Para poder buscar en el historial commits que contengan cierto texto en el mensaje usamos el comando:

git log --grep=”palabra_clave”

• ¿Cómo borrar el historial de Git?

No es fácil borrar el historial, pero podés eliminar un commit reciente usando:

git reset --hard HEAD~n (donde n es el número de commits a retroceder).

Ó

git rebase -i HEAD~n

• ¿Qué es un repositorio privado en GitHub?

Es un repositorio que solo pueden ver y acceder las personas que vos autorices.

• ¿Cómo crear un repositorio privado en GitHub?

Al momento de crear un nuevo repositorio en GitHub, en el formulario, te da la opción que elijas si querés que el repositorio sea privado o público, simplemente marcas privado y listo.

• ¿Cómo invitar a alguien a un repositorio privado en GitHub?

1) Vas a la pestaña “Settings” del repositorio.
2) Haces click en “Collaborators”.
3) En la sección “Manage access” haces click en el botón “Add people”.
4) Agrega al usuario con su nombre de GitHub.

• ¿Qué es un repositorio público en GitHub?

Es un repositorio visible para cualquiera, incluso sin iniciar sesión. Cualquiera lo puede ver y clonar, es accesible para todo el mundo.

• ¿Cómo crear un repositorio público en GitHub?

Al momento de crear un nuevo repositorio en GitHub, en el formulario, te da la opción que elijas si querés que el repositorio sea privado o público, simplemente marcas público y listo.

• ¿Cómo compartir un repositorio público en GitHub?

Copiás la URL del repositorio desde la barra del navegador y la compartís. Cualquiera con el link puede acceder.